REMARKS

Claims 9, 11, 12, 18, and 19 have been amended. Claims 1-31 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 112, Second Paragraph, Rejection:

The Examiner rejected claims 1-30 under 35 U.S.C. § 112, second paragraph, as indefinite. Specifically, the Examiner submits that the amendments to each of the independent claims requires that the index be "fixed", but that an index value derived from an address is variable (as described in paragraph 47 of Applicants' specification). The Examiner states that he has chosen to simply view the word "fixed" as indefinite. The Examiner has apparently misread the claims. For reasons explained in more detail below, Applicants assert that the claims are not indefinite, and that it is improper for the Examiner to ignore the specific wording of Applicants' claims.

First, the Examiner has misinterpreted Applicants' amended claims. The claims recite, "*the indexed STLF buffer includes a plurality of entries each of which is selectable using a fixed index value unique to that entry*." In other words, each location in the buffer is selectable using a fixed index value. The index value used to select any particular location in the buffer does not depend on the contents of that location. This is a clear distinction over the cited art, as described in more detail below. Other limitations of the claims require that in order to determine which of the locations to select (and to access), an index value is generated dependent on an address (as noted by the Examiner), and that this generated index value matches one of these fixed index values. Therefore, the location that is selected by the generated index value is the one location whose fixed index value matches the generated index value, regardless of its contents. As noted in the previous Response, Applicants' amendment was made to clarify this indexing operation of Applicants' claimed invention, in which the generated index value is not merely "used to select" an entry, but that it is used to "index into the indexed STLF buffer," i.e., to select one of the entries in the STLF buffer according to the entry's unique fixed index value.

Applicants further assert that this amendment is clearly supported in Applicants' specification. For example, paragraph [0034] states, in part, "Each entry 320 in the STLF buffer 305 may be selected by a unique index. The load store unit 126 may generate an index from all or part of the address of an operation and use the index to select which entry to access for that operation" (emphasis added). In another example, paragraph [0035] states, in part, "At that point, the load store unit may allocate an entry 320 in the STLF buffer 305 to the store operation. The load store unit 126 may select which entry 320 to allocate to a store operation by calculating an index from at least a portion of the store operation's address. The entry 320 selected by that index may be allocated to the store operation" (emphasis added). Similarly, paragraph [0036] includes the following, "Whenever a load operation's address becomes available, the load store unit 126 may generate an index based on all or part of the load's address in order to select an entry 320 within the STLF buffer 305" (emphasis added), and paragraph [0037] includes this passage, "Accordingly, whenever a load operation's address is used to index into the STLF buffer 305, a single entry 320 will be selected" (emphasis added). Paragraphs [0038] - [0041] describe specific examples of generating an index value and using it to select particular entries in an STLF buffer that includes sixteen locations indexed by the fixed values 0x0h through 0xFh. Applicants submit, therefore, that Applicants' specification clearly supports the above-referenced amendment.

For at least the reasons above, Applicants submit that the Examiner's rejection under 35 U.S.C. § 112, second paragraph, is improper and respectfully request the removal thereof.

Section 102(b) Rejections:

The Examiner rejected claims 1-3, 5, 8-10, 13-16, 18, 20-22, 24, 27, 28 and 31 under 35 U.S.C. § 102(b) as being anticipated by Hughes WO, and claims 1-3, 5, 8-10, 13-16, 18, 20-22, 24, 27, 28 and 31 under 35 U.S.C. § 102(b) as being anticipated by Webb et al. (U.S. Patent 6,360,314) (hereinafter "Webb"). Applicants traverse these rejections for at least the following reasons.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner disagreed with Applicants' arguments that merely using a value to select an entry cannot be considered "indexing", and provides a definition of the noun "index". Applicants noted that the Examiner has referred to the noun "index", not the verb "indexing". Applicants maintained their traversal of the rejection, but amended claim 1 (and independent claims 14 and 20) to clarify both the generated index value and the indexing operation of Applicants' claimed invention. **Applicants note that the Examiner ignored the amendment of these claims in repeating his rejections from the previous Office Action.** As discussed below, the cited art clearly fails to disclose the limitations of Applicants' claims involving this generated index value and indexing operation.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner disagreed with Applicants' arguments regarding "generating an index" and submits that, as interpreted in this Office Action, an index can be generated in at least two ways: 1) When the bits used for the index are determined based on the output of any logic components or 2) taking the pre-existing bits and organizing them in a fashion that they can be utilized as an index. Applicants asserted, however, that the Examiner did not cite anything in the art of record supporting these interpretations. **Applicants also assert that, as discussed below, the cited art does not teach <u>all the limitations</u> of Applicants' claims, even if this interpretation of "generating an index" is applied to the cited art.**

<u>First ground of rejection under 35 U.S.C. § 102(b):</u>

The Examiner rejected claims 1-3, 5, 8-10, 13-16, 18, 20-22, 24, 27, 28 and 31 under 35 U.S.C. § 102(b) as being anticipated by Hughes WO. Applicants traverse this rejection for at least the following reasons.

Regarding claim 1, contrary to the Examiner's assertion, Hughes fails to teach or suggest an indexed STLF (Store-to-Load Forwarding) buffer including *a plurality of entries <u>each of which is selectable using a fixed index value unique to that entry</u>* and a load store unit configured to *generate an index value dependent on at least a portion of an address of a load operation, wherein the <u>generated index value is one of the fixed index values</u>; <u>index into the indexed STLF buffer using the generated index value</u> to select one of the plurality of entries; and forward data included in the one of the plurality of entries selected by the generated index value as a result of the load operation.*

The generated index value of Applicants' invention is not merely "used to select" an entry, but that it is used to "index into the indexed STLF buffer." Claim 1, as amended, clarifies this limitation by reciting that the indexed STLP buffer includes a plurality of entries <u>each of which is selectable using a fixed index value unique to that entry</u>, and that the <u>generated index value is one of the fixed index values</u>. Applicants assert that as amended, the independent claims clearly distinguish over the cited art, since the cited reference uses an index portion of an address in a <u>comparison operation</u> to select an entry in a buffer that uses a <u>variable value</u>, rather than a fixed value, to select each entry in the buffer. The Examiner's citations describe that a comparison is made between an index portion of a load address and an index portion of an entry in an LS2 buffer (or store queue) to determine whether to forward data from the store queue. Applicants assert that there is nothing in Hughes that teaches or suggests that <u>each entry in the buffer is selectable using a fixed index value unique to the entry</u>, nor that a generated index value matching this fixed index value is used to select an entry in the buffer. **Instead, the entries in the LS2 buffer of Hughes are selectable <u>based on the contents of the</u>**

entry, (i.e., an index portion of the store address contained in the entry) which is clearly a variable value.

In the Response to Arguments section of the Office Action of May 18, 2007, the Examiner discusses the amendment to the independent claims requiring that the index value be fixed. As discussed above regarding the rejection under 35 U.S.C. § 112, second paragraph, **the Examiner has misinterpreted the amendment to the independent claims.** As discussed in detail above, the amendment to claim 1, for example, requires that each entry in the STLF buffer is selectable using a fixed index value unique to that entry (rather than one that is dependent on the contents of the entry). The remaining limitations of claim 1 recite generating an index value that matches one of these fixed values, and selecting the entry selectable using that fixed value. These limitations are clearly not taught by Hughes. As discussed above, the entries in the LS2 buffer of Hughes are selectable based on their contents, and their contents are clearly variable, not fixed.

Therefore, for at least the reasons above, the rejection of claim 1 is not supported by the cited art and Applicants respectfully request the withdrawal thereof.

Independent claims 14 and 20 each include limitations similar to those discussed above regarding claim 1. Therefore, the arguments presented above regarding claim 1 apply with equal force to these claims as well.

Regarding claim 5, contrary to the Examiner's assertion, Hughes fails to teach or suggest *the load store unit is configured to select which one of the plurality of entries to allocate to a store operation by generating an additional index value dependent on at least a portion of an address of the store operation*. The Examiner cited FIG. 1 (ADDR – Tag) as teaching this limitation. In the Request for Continuing Examination, Applicants asserted that the address tag portion of an entry in store queue 400 is not described as an additional index value used to select which one of the plurality of entries (i.e., to select which one of the plurality of entries to **allocate to a store operation**.) In his

Response, the Examiner cited Hughes, page 7, line 10-14 as disclosing that the tag is used in "selecting an entry from the STLF buffer." However, this passage describes the <u>load address</u>, when it may be available for comparison, when the load's hit signal may be determined, and when the way indication for the load may be determined. **It has absolutely nothing to do with selecting an entry <u>to allocate to a store operation</u>**, (i.e., to select an entry in which to store information about a store operation). Applicants assert that nothing in Hughes discloses the additional index value of claim 5, i.e., an index value generated dependent on at least a portion of an address of the store operation and used to select which one of the plurality of STLF entries (or a store queue entry) is allocated to a store operation.

In the Response to Arguments section of the Office Action of May 18, 2007, the Examiner submits that Hughes allocates memory using a two way set associative structure (page 16, lines 6-8) and that this satisfies the claim limitations of claim 5. This passage merely states that data cache 28 has a capacity of storing up to 64 kilobytes of data in a two way set associative structure. **It teaches nothing about how a particular entry in the cache is <u>selected for allocation</u> to a given store operation.** Applicants assert that a two way set associative structure does not inherently allocate entries in the manner recited in claim 5, but instead affects the way previously stored entries are retrieved.

For at least the reasons above, the rejection of claim 5 is unsupported by the cited art and removal thereof is respectfully requested. Claim 16 and 24 include limitations similar to claim 5, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 8, contrary to the Examiner's assertion, Hughes fails to teach or suggest *the additional index value comprises a portion of the address targeted by the store operation*. The Examiner again cited FIG. 1 (ADDR – tag) as teaching this limitation. However, as discussed above, Hughes does not teach generating the additional index value of claim 5 at all, much less one comprising a portion of the address targeted by the store operation. The address tag in FIG. 1 is <u>the portion of the store address</u> that is stored as a tag by the data cache, not <u>an additional index value used in selecting an entry in a STLF buffer to allocate to a store operation</u>, as required by Applicants' claims 5 and 8.

In the Response to Arguments section of the Office Action of May 18, 2007, the Examiner submits that claim 8 fails to further limit claim 5 besides requiring that the additional index depend on the address "targeted by" the store operation, rather than "of" the store operation. Applicants disagree and note that claim 8 does not recite that the additional index value merely "depends on" the address targeted by the store operation, but that it actually <u>comprises a portion of</u> this address. This is clearly an additional limitation over claim 5. Applicants assert that this additional limitation is also not taught by Hughes.

For at least the reasons above, the rejection of claim 8 is unsupported by the cited art and removal thereof is respectfully requested. Claim 27 includes limitations similar to claim 8, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 10, contrary to the Examiner's assertion, Hughes fails to teach or suggest *the STLF checker is configured to perform an associative address comparison to identify all issued store operations targeting a same address as the load operation and to implement a find-first algorithm to select a youngest issued store operation that is older than the load operation*. In the Final Action, the Examiner cited page 8, lines 14-17 as teaching these limitations. However, while this passage states, "Hit control circuit 402 may determine the youngest (most recently executed) store in program order among the stores corresponding to entries which are hit and may forward the data from that entry," there is nothing in Hughes that discloses the specific limitations of claim 10. For example, there is no mention of <u>implementing a find-first algorithm</u>, as recited in claim 10. Therefore Hughes cannot be said to anticipate claim 10.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner disagrees with Applicants' arguments above, stating, "It is unclear what definition Applicant believes the claimed invention requires from the term "find-first algorithm", but it appears sufficient to say that the remainder of the functionality of claim 10 can be considered a "find-first algorithm" and Applicant says nothing to suspect the contrary." Applicants again submit that while the Examiner's passage describes a similar goal as that recited in claim 10, "to select a youngest issued store operation that is older than the load operation", it does not describe <u>how</u> this determination is implemented. Claim 10 recites that this is implemented using a find-first algorithm. Hughes, however is silent as to the specifics of the determination. Applicants assert that many different methods may be implemented to select the youngest of the operations, depending on the order in which the entries are stored (e.g., whether they are stored in program order or execution order), what additional information may be stored within each entry (e.g., an indicator of age, or program order), or other implementation details. Therefore, the "youngest" of the operations may not be the "first" of the operations that correspond to a hit, and may not be selectable using a "find-first algorithm." The Examiner's other cited reference, for example, uses an INUM field to indicate the relative age of various entries in a store queue.

In the Response to Arguments section of the Office Action of May 18, 2007, the Examiner submits that it is unclear what weight to give the term "find-first" algorithm. The Examiner presumes that a "find-first" algorithm is merely an algorithm "to select a youngest issued store operation that is older than the load operation." Applicants assert that the term "find-first" algorithm is well known in the computer arts. For example, it is common for a library of programming utilities to include a routine that finds the first occurrence of a string within an input. In another example, various microprocessors include an instruction that finds the first occurrence of a zero or non-zero value within a byte or word. At the very least, in order for Hughes to anticipate the limitations of claim 10, there must be some mention of an algorithm for searching (and finding) a first occurrence of an issued store operation from among those having a same address as a load operation in the system of Hughes. There is nothing in Hughes that describes that this "hit control circuit 402" implements such a search algorithm. Therefore, Hughes cannot be said to anticipate claim 10.

For at least the reasons above, the rejection of claim 10 is unsupported by the cited art and removal thereof is respectfully requested. Claim 28 includes limitations similar to claim 10, and so the arguments presented above apply with equal force to this claim, as well.

Second ground of rejection under 35 U.S.C. § 102(b):

The Examiner rejected claims 1-3, 5, 8-10, 13-16, 18, 20-22, 24, 27, 28 and 31 under 35 U.S.C. § 102(b) as being anticipated by Webb, et al. (U.S. Patent 6,360,314) (hereinafter "Webb"). Applicants traverse this rejection for at least the following reasons.

Regarding claim 1, contrary to the Examiner's assertion, Webb fails to teach or suggest an indexed STLF (Store-to-Load Forwarding) buffer including *a plurality of entries_each of which is selectable using a fixed index value unique to that entry* and a load store unit configured to *generate an index value dependent on at least a portion of an address of a load operation, wherein the generated index value is one of the fixed index values; index into the indexed STLF buffer using the generated index value to select one of the plurality of entries; and forward data included in the one of the plurality of entries selected by the generated index value as a result of the load operation.*

First, as noted in the Request for Continuing Examination, Webb does not disclose a load store unit configured to store information associated with load and store operations, nor including an indexed Store-to-Load Forwarding (STLF) buffer having the limitations of claim 1. The Examiner refers to FIG. 4, including buffer 428 and queue 426 as teaching these limitations. However, FIG. 4 clearly illustrates that store data buffer 428 and store queue 426 are components of data cache subsystem 420, and not load/store unit 418. In addition, these is nothing in Webb describing that data included in the entry selected by the generated index value is forwarded as a result of the load operation, as required by claim 1. Instead, data from store data buffer 428, which is not selected using the generated index value to index into the store data buffer, is forwarded as a result of a load operation in which the tag in the store queue and the tag reference match for a particular entry in the store queue. In Webb, this store queue entry, not a store data buffer entry, may be selected based on an address comparison between an address of an issuing load and a store address in the store queue entry. There is nothing that teaches a store data buffer entry is selected based on a generated index value indexing into the store data buffer, as recited in claim 1.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner submits that the STLF buffer and associated logic constitute part of the load/store unit as they all assist in performing load and store operations. Applicants assert, however, that even if store data buffer 428 and store queue 426 were considered part of the load/store unit, they do not teach or suggest all the limitations of the STLF buffer recited in claim 1, specifically that each entry of the STLF buffer is selectable using a fixed index value unique to that entry, using a generated index value to index into the STLF buffer to select an entry, and forwarding data included in the entry as a result of the load. Instead, Webb discloses that data is forwarded from the store data buffer if the tag in the selected store queue entry and the tag reference match for a particular store queue entry.

Further regarding claim 1, Webb does not disclose indexing into the indexed STLF buffer using the generated index value to select one of the entries in the STLF buffer and forwarding data from the selected entry. The Examiner previously cited column 5, lines 5-8 and 19-22, as teaching these limitations. However, there is nothing in these citations that teaches that the load store unit, or any other apparatus, is configured to generate an index value used to index into an indexed STLF buffer, or to Webb's store data buffer 428, which the Examiner equates with Applicants' STLF buffer. Instead, these citations describe indexing into dcache unit 430, and one of its component (tag store 432.)

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner submitted that Webb also describes indexing into the STL buffer as well, in column 6, lines 6-10. Applicants assert that this passage also does not describe indexing into store data buffer 428 or store queue 426. This passage states, in part, "the tag store 432 and the store queue 426 are queried using bits 14:0 of the virtual address for the virtual index and bits 43:13 of the translated physical address for the tag comparison. If there is a match in the store queue 426 at block 510, then corresponding data is read from the store data buffer 428 at block 512." However, this "virtual index" is not used to index into a buffer whose entries are selectable by indexing into the buffer using a fixed index value to select an entry. Instead, this virtual index is used in an <u>address comparison</u> to select a matching entry.

In the Response to Arguments section of the Office Action of May 18, 2007, the Examiner submits that the STLF buffer is anticipated by the combination of buffer 428 and queue 426. The Examiner further submits that "the plurality of entries" are included in the STLF buffer, which is satisfied by 428, and that data is forwarded from "one of the plurality of entries" as described in column 2, lines 8-15. However, the Examiner's own citation in column 2, lines 8-15, describes the operation as follows, "An address of an issuing load is <u>compared against the store queue address for each store queue entry</u>. In response to an address match between the issuing load and a particular store queue entry, the store data entry in the store data buffer that corresponds to the particular store queue entry (referred to as the "address-matching store queue entry") is passed to the execution unit when the issuing load is younger in program order than the address-matching store queue entry." Thus, it is clear that an address-matching operation, not an indexing operation such as the one recited in claim 1, is used to select an entry in store queue 426. If there is a match, a corresponding entry in the store data buffer 428 is forwarded. This clearly does not teach the limitations of Applicants' claim 1.

For at least the reasons above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Independent claims 14 and 20 include limitations similar to those discussed above regarding claim 1, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 5, contrary to the Examiner's assertion, Webb fails to teach or suggest *the load store unit is configured to select which one of the plurality of entries to allocate to a store operation by generating an additional index value dependent on at least a portion of an address of the store operation.* The Examiner previously cited column 5, lines 5-8 and 19-23 as teaching this limitation. However, these passages does not describe generating <u>an additional index value</u> that is used to select an entry to allocate to a store operation (or any two indices, for that matter). These two passages refer to <u>the same virtual index</u>, used to index into dcache unit 430 (specifically tag store 432) and store queue 426, as discussed above.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner submits that the additional index value as claimed "is required to depend on a portion of the address of the store operation (col. 5 lines 5-8 and 19-23) and data operated on by the store operation (col. 6 lines 51-59)." Applicants disagree. Claim 5 does not require <u>the additional index value</u> to be dependent on anything other "at least a portion of an address of the store operation." The Examiner goes on to state that the combination of this data is considered to be "an additional index". This statement is completely unsupported in the cited art. The Examiner's citation in column 6 teaches that in Webb, additional fields may be compared to determine a hit (e.g., a size field and/or an INUM field.) Neither of these constitutes <u>an index value</u> at all, much less <u>an additional index value for use in selecting an entry to allocate to a store operation</u>, and <u>neither is dependent on a portion of an address, as required by claim 5</u>. Applicants assert, therefore, that this passage has nothing to do with the additional index value of Applicants' claim 5.

In the Response to Arguments section of the Office Action of May 18, 2007, the Examiner submits that the indexing operation referred to in claim 5 appears to be an indexing technique that is broad enough to fall under several techniques. The Examiner further submits that this includes the "set associative" technique used in Webb (column 4, line 58). Applicants assert, however, that the Examiner is ignoring the specific wording of claim 5. Claim 5 recites that *the load store unit is configured* *<u>to select which one of the plurality of entries to allocate to a store operation</u>* *by generating an additional index value dependent on at least a portion of an address of the store operation.* The cited passage refers generally to the dcache of Webb (not to store queue 426 or store data buffer 428) and merely states, "In the preferred embodiment, the dcache unit 430 is a 64 KB, 2-way set-associative, virtually-indexed and physically-tagged data cache." **This has absolutely nothing to do with <u>selection of an entry in a STLF buffer to allocate to a given store operation</u>.** Therefore, Webb cannot be said to anticipate claim 5.

For at least the reasons above, the rejection of claim 5 is not supported by the cited art and removal thereof is respectfully requested. Claims 16 and 24 include limitations similar to claim 5, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 8, contrary to the Examiner's assertion, Webb fails to teach or suggest *the additional index value comprises a portion of the address targeted by the store operation.* The Examiner again cites column 5, lines 19-22, as teaching this limitation. However, this citation describes indexing (using a portion of an address) into <u>dcache unit 430</u>, not into <u>a STLF buffer</u>, or into Webb's store queue 426 and store data buffer 428. Using an address to index into dcache unit 430 teaches nothing about the composition of <u>the additional index value</u> (for selecting an entry in the STLF buffer) referred to in Applicants' claim 8.

In the Response to Arguments section, the Examiner "points to FIG. 4 indicating that the additional index value generated by the load/store unit is connected to the buffer store queue 426 in addition to dcache 430, as indicated by lines 442 and 446." However, FIG. 4 illustrates that the <u>same index</u> as the one the Examiner refers to in his rejection of claim 1 (described in column 5, lines 5-8 and 19-22) is connected to both the dcache and the buffer store queue. Therefore, this cannot be considered "an additional index value" as the Examiner suggests.

Applicants note that the Examiner failed to address Applicants' arguments above in the Office Action mailed November 28, 2006, or the Office Action mailed May 18, 2007.

For at least the reasons above, the rejection of claim 8 is not supported by the cited art and removal thereof is respectfully requested. Claim 27 includes limitations similar to claim 8, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 9, contrary to the Examiner's assertion, Webb fails to teach or suggest *the load store unit further comprises a STLF checker configured to verify correct operation of the STLF buffer in said forwarding data as a result of the load operation.* Applicants traverse this rejection for at least the reasons given below regarding the rejection of claim 9 under 35 U.S.C. § 103(a) as being unpatentable over Webb in view of common art. Similar remarks apply also to claim 18.

Section 103(a) Rejections:

The Examiner rejected claims 9 and 18 under 35 U.S.C. § 103(a) as being unpatentable over Webb in view of common art, claims 11, 12, 19, 29 and 30 as being unpatentable over Webb in view common art and Hughes in view of common art, and claims 4 and 23 as being unpatentable over Hughes (WO 01/35212) (hereinafter "Hughes WO" and Webb. Applicants traverse these rejections for at least the following reasons.

<u>First ground of rejection under 35 U.S.C. § 103(a):</u>

The Examiner rejected claims 9 and 18 under 35 U.S.C. § 103(a) as being unpatentable over Webb in view of common art. Applicants traverse this rejection for at least the following reasons.

Regarding claim 9, contrary to the Examiner's assertion, Webb fails to teach or suggest *the load store unit further comprises a STLF checker configured to verify correct operation of the STLF buffer in said forwarding data as a result of the load operation.* In the Response to Arguments section of the Final Action of June 2, 2006, the Examiner submitted, "there must inherently exist logic to verify that the correct entry is forwarded when multiple stores are pending for the same address location. If the youngest store is not forwarded upon a subsequent load, incorrect operation may result." Applicants noted that the <u>data bypass method and apparatus</u> described at the Examiner's citation in Webb includes logic to "attempt to <u>provide the data of the most recent store</u> to a subsequently issuing load and thereby avoid getting older data to the load," (column 7, lines 18-20). Attempting to provide the correct data is clearly not equivalent to the limitation of Applicants' claim 9, which recites *the load store unit further comprises <u>a STLF checker</u> configured to <u>verify correct operation of the STLF buffer in said forwarding data as a result of the load operation</u>.* Furthermore, Applicants remind the Examiner that "To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is **<u>necessarily present</u>** in the things described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.'" *In re Robertson*, 169 F.3d 743, 745, 49 USPA2d 1949, 1950-51 (Fed. Cir. 1999) (emphasis added). The Examiner has not cited anything in Webb that teaches or suggests *the load store unit further comprises <u>a STLF checker</u> configured to <u>verify operation of the STLF buffer</u>.*

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner again asserted that the use of inherency is correct. The Examiner states, "The functionality considered to be inherent is required for proper functionality of the invention as disclosed. If incorrect entries were forwarded, the processor would be broken and unable to function as the reference states.... There must be logic to insure proper verification." Applicants disagreed and argued that the invention of Webb could be implemented in such a way that incorrect entries cannot be forwarded (e.g., in such a way as to not forward an entry unless its correct selection is deterministic). In such embodiments, there would be no need to verify this operation.

In the Office Action mailed May 18, 2007, the Examiner states, "Examiner asserts that Applicant's proposed technique still requires a verification of the operation. If Webb chooses not to forward an entry unless it is correct, there must be some way to verify whether it is correct." The Examiner has misunderstood Applicants' argument. Applicants' argument is that the invention of Webb could be implemented such that entries are <u>not ever forwarded</u> until it is possible to determine that the entry should be forwarded (rather than speculatively forwarding an entry before there is enough information available to determine that it contains data that should be forwarded). By waiting until the selection is deterministic, the forwarding operation would always be correct and would not need to be verified. By contrast, Applicants' claims recite forwarding an entry and then verifying that whether it was correct in forwarding the data, using an STLF checker. This is clearly not taught by Webb and would not be inherent in the system of Webb.

In the Office Action of May 18, 2007, the Examiner also states, "However, even if this inherency were not true, it is further obvious to use a STLF checker to verify the operation of the STLF buffer. Examiner takes Office Notice that accurate storing of information is typically required for accurate output. Examiner further asserts that errors can occur based on hazards, soft errors, etc. Therefore, it would have been obvious at the time of the invention for one of ordinary skill in the art to take the processor of Webb and allow for a STLF checker to verify correct operation on the STLF buffer." **Applicants assert that claim 9 has nothing to do with "accurate storing of information" in the STLF buffer, but with "a STLF checker configured to verify correct operation of**

the STLF buffer <u>in said forwarding data</u> as a result of the load operation." The error types noted by the Examiner have nothing to do with the <u>correctness of a forwarding operation</u> of an STLF buffer. Therefore, the Examiner has again failed to provide sufficient motivation to modify the system of Webb or to show how Webb teaches or suggests the limitations of claim 9.

For at least the reasons above, the rejection of claim 9 is not supported by the cited art and removal thereof is respectfully requested. Claim 18 includes limitations similar to claim 9, and so the arguments presented above apply with equal force to this claim, as well.

<u>Second ground of rejection under 35 U.S.C. § 103(a):</u>

The Examiner rejected claims 11, 12, 19, 29 and 30 under 35 U.S.C. § 103(a) as being unpatentable over Webb in view common art and Hughes in view of common art. Applicants traverse this rejection for at least the following reasons.

Regarding claim 11, contrary to the Examiner's assertion, Webb fails to teach or suggest *the STLF checker is configured to replay the load operation if the STLF checker identifies incorrect operation of the STLF buffer*. The Examiner previously cited column 1, lines 34-38 as teaching this limitation and asserts that if a load operation does not result in data being provided from the cache, the memory load is replayed to the main memory. This is incorrect. First, this citation teaches nothing about <u>replaying a load operation</u>. It merely describes that the data for a load is <u>retrieved from main memory</u>, instead of from the cache, if it is not found in the cache. Furthermore, this citation has <u>nothing to do with</u> a STLF checker identifying <u>incorrect operation of the STLF buffer</u>. In fact, this citation has nothing to do with the <u>STLF buffer</u> at all. Instead, as stated above, this citation describes only that if <u>data is not found in the cache</u>, it is <u>retrieved from main memory</u>.

Applicants noted that the Examiner failed to address Applicants' argument above that this citation has nothing to do with incorrect operation of an STLF buffer in the Office Action mailed November 28, 2006. In fact, this citation specifically describes the operation of a data cache in a <u>prior art system</u>, i.e., in <u>one that does not include the bypass mechanism of Webb's invention</u>. Therefore, it cannot teach anything about an STLF checker replaying an operation in response to identifying incorrect operation of the STLF buffer, as required by claim 11, as no such buffer exists.

In the Response to Arguments section of the Final Action of June 2, 2006, the Examiner submitted, "when a load misses after a first attempt in the buffer of Webb's system, the load must be replayed (attempted again) to another memory device (in this case, the dcache unit 430). There is nothing is applicant's specification defining the term "replay" and hence, it has been awarded its broadest reasonable definition." Applicants disagreed with the Examiner's interpretation of the term "replay" and also with his assertion that Applicants' specification does not include anything defining this term. Applicants noted that the specification includes a specific example of how a replay may be caused by the STLF checker on page 16, lines 17-18, "The STLF checker 303 may cancel load operations that either incorrectly forwarded in STLF buffer 305 or which incorrectly did not forward in STLF buffer 305. In such situations, the STLF checker 303 may cause the load operation to be replayed (e.g., by providing a signal to the scheduler 118)." Thus, it is clear from Applicants' specification that the term "replay" was intended to refer to situations in which a load operation is canceled and then <u>rescheduled for execution</u> (i.e., the current instance of the load operation is cancelled and another instance of the load operation is scheduled for execution) rather than to situations in which <u>a current load operation</u> may search more than one level of memory in an attempt to fulfill the load.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner submitted that the cited portion of the specification does not provide a suitable definition of the term "replay". The Examiner submitted that when the specification says, "e.g., by providing a signal to the scheduler 118", this is referring to an example of how the word replay may be used, not to a particular definition for the word. Applicants disagree and also note that the Examiner has not cited any evidence in the art of record that his own interpretation of the word "replay" is a reasonable one.

In addition, **Applicants asserted that retrieving data for a load from main memory after a cache miss does not inherently require <u>replaying</u> a load operation (or attempting it again) as the Examiner suggests.** Instead, many processors having a cache initiate an access main memory in parallel with an access in the cache, so that if the data is not found in the cache, retrieval from main memory has already been initiated and is not delayed until a hit or miss indication is received. In these cases, the load operation would clearly not have to be replayed (or attempted again). It would have already been initiated and would just be allowed to complete.

In the Office Action mailed May 18, 2007, the Examiner admits that Hughes and Webb fail to disclose reissuing instructions that result in incorrect output and instructions dependent on those instructions. The Examiner also states that claim 11 is more appropriately rejected as an obviousness rejection (see the Response to Arguments section). The Examiner then repeats his arguments of the previous Office Actions regarding claim 12, "Examiner asserts that it is common in the art to reissue these types of instructions when the output is not a desired output. Hughes and Web *[sic]* would have been motivated to utilize this technique because reissuing instructions is a common, simple, fast, and effective technique for gaining the correct output. In fact, the Examiner is not aware of any other technique. It would have been obvious at the time of the invention for one of ordinary skill in the art to take the processing system of either Hughes and Web *[sic]* and combine it with the ability to reissue instructions when the original output is not a desired result."

In previous remarks regarding claim 12, the Examiner submitted that Hughes and Webb disclose the microprocessor of claim 9 and the method of claim 28. The Examiner further submits that he believes it is inherent that a program must <u>reissue at least some element of the previous instruction to receive a desired output</u>, even though this is not expressed explicitly in either reference (emphasis added). Applicants disagreed with the Examiner's characterization that reissuing instructions is a "common, simple, fast, and effective technique for gaining the correct output" and noted that the Examiner has not provided any evidence of record to teach or suggest that this is the case. **Applicants again assert that the Examiner has not provided such evidence or explained how he believes an STLF checker that operates according to Applicants' claims could be incorporated into the system of either Webb or Hughes.** For example, since neither Webb nor Hughes discloses the STLF checker of Applicants' claims, neither one can teach or suggest the additional limitation that <u>*the STLF checker*</u> *is configured to replay the load operation* <u>*in response to the STLF checker identifying incorrect operation of the STLF buffer*</u>.

For at least the reasons above, the rejection of claim 11 is unsupported by the cited art and removal thereof is respectfully requested. Claims 19 and 29 include limitations similar to claim 11, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 12, contrary to the Examiner's assertion, the cited references, in view of common art, fail to teach or suggest *the STLF checker is configured to replay one or more additional operations that are dependent on the load operation in response to the STLF checker detecting incorrect operation of the STLF buffer*. As noted above, the Examiner previously submitted that Hughes and Webb disclose the microprocessor of claim 9 and the method of claim 28 and that he believes it is inherent that a program must <u>reissue at least some element of the previous instruction to receive a desired output</u>, even though this is not expressed explicitly in either reference (emphasis added). The Examiner asserted that it is common in the art to reissue instructions when the output is not a desired result and that Hughes and Webb would have been motivated to utilize this technique because reissuing instructions is a common, simple, fast, and effective technique for gaining the correct output. The Examiner states that he is not aware of any other technique. The Examiner submitted, therefore, that it would have been obvious at the time of the invention for one of ordinary skill in the art to take the processing system of either Hughes or Webb and combine it with the ability to reissue instructions when the original output is not a desired result.

Applicants traversed the Examiner's statements and the Examiner's reliance on allegedly common art. First, Applicants note that claim 12 does not recite reissuing <u>the previous instruction</u> (or at least some element of the previous instruction) in response to the output being an incorrect result, as the Examiner suggests. Instead, it recites that the STLF checker is configured to replay <u>one or more additional operations that are dependent on a load operation</u> for which the STLF buffer operated incorrectly. Applicants assert that merely reissuing the previous instruction when the output is not a desired result does not teach or suggest the limitation referenced above. Applicants again submit that there is nothing <u>**inherent**</u> about replaying <u>additional, dependent instructions</u> when the previous instruction is reissued, and that there are many ways a system could operate correctly without replaying the dependent instructions. For example, in some systems, any dependent instructions may be prevented from executing until correct operation of an STLF buffer, or a similar mechanism, has been verified for the load operation on which they depend. Therefore, there would be no need to replay the <u>dependent instructions</u> following incorrect operation of the buffer with respect to the load operation.

In the Response to Arguments section of the Office Action of May 18, 2007, the Examiner admits that this alternative would prevent the requirement of reissuing dependent instructions. The Examiner notes that this is no longer an inherency rejection, but an obviousness rejection. The Examiner submits, 'Applicant's solution would severely cripple out-of-order processing to a point that most efficiency derived from this several decade old technique would be lost. The technique of reissuing dependent instructions (including "the previous instruction") is overtly obvious to anyone skilled in the art and is doubtlessly "common, simple, fast, and effective" compared to Applicant's alternative solution.'

Applicants again disagree with the Examiner's characterization that reissuing instructions is a "common, simple, fast, and effective technique for gaining the correct output" and note that the Examiner has still not provided any evidence of record to teach or suggest that this is the case, or any explanation of how such a technique could be incorporated into the systems of Hughes or Webb to teach the limitations of claim 12. As noted above regarding claim 11, since neither Webb nor Hughes discloses the STLF checker of Applicants' claims, neither one can teach or suggest the additional limitation that *<u>the STLF checker</u> is configured to replay one or more additional operations that are dependent on the load operation<u> in response to the STLF checker detecting incorrect operation</u> of the STLF buffer.*

For at least the reasons above, Applicants assert that the rejection of claims 12 and 30 are not supported by the cited art, and removal thereof is respectfully requested.

<u>**Third ground of rejection under 35 U.S.C. § 103(a):**</u>

The Examiner rejected claims 4 and 23 under 35 U.S.C. § 103(a) as being unpatentable over Hughes (WO 01/35212) (hereinafter "Hughes WO" and Webb. Applicants traverse this rejection for at least the following reasons.

Regarding claim 4, contrary to the Examiner's assertion, Hughes and Webb fail to teach or suggest *each of the plurality of entries in the STLF buffer has a capacity to store a maximum amount of data that can be written by a store operation*. **The Examiner has admitted, in previous remarks, that both Hughes and Webb fail to disclose this limitation.**

The Examiner previously noted, "Hughes does not disclose the data bus width of the processor of his invention." In the Final Action of June 2, 2006, the Examiner took Official Notice that data buses of size 8, 16, 32, or 64 bits are extremely well known in the art and that with any of these data buses in place in Hughes invention, the data buffer would be able to hold at least the maximum amount of data specified by a data store operation. Since the Examiner has admitted that Hughes does not disclose the data bus width of the processor (or mention anything about the width of entries in the buffers of Hughes), Applicants assert that nothing in Hughes teaches this limitation of claim 4. Applicants previously asserted that the bus width is irrelevant, and notes that claim 4 recites nothing about a bus width. Applicants also note that the Examiner states, "the data buffer would be able to hold at least the maximum amount of data specified by a data store operation" (emphasis added). Applicants' claim, however, recites that each entry in the buffer has a capacity to store a maximum amount of data that can be written by a store operation. Applicants again assert that the maximum amount of data that can be written by a store operation is not necessarily the same as, or dependent on, the data bus width, but instead may be dependent on the instruction set architecture (i.e., the instruction sets of different processors may provide for store operations involving different numbers of bytes of data and/or the implementation of the instruction set may provide for different numbers of bus cycles for each store operation). **Therefore, Applicants asserted that the Examiner's Official Notice does not teach or suggest the limitations of claim 4.**

In the Response to Arguments section of the Office Action of May 18, 2007, the Examiner submits that there is nothing within the reference to indicate the use of multi-clock memory operations and cites page 25, line 30 to page 26, line 5. This passage describes the control circuitry of the LS2 buffer of Hughes selecting which memory operations to reprobe in data cache 28. **It has nothing to do with the number of clock cycles used for memory operations, the size of each entry in the LS2 buffer, or the size of each buffer entry matching the data bus width, as suggested by the Examiner.** The Examiner further submits, "Additionally, even if there were multi-clock memory operations, it is still a reasonable interpretation to consider the operation of each separate clock cycle to be a separate operation." Applicants respectfully disagree. Applicants note the plain language of the claim recites *each of the plurality of entries in the STLF buffer has a capacity to store a maximum amount of data that can be written by a store operation*. **Applicants again assert that store operations in various processor**

architectures are clearly capable of writing different amounts of data and nothing in Hughes discloses this limitation of claim 4, as previously admitted by the Examiner.

As noted above, the Examiner has admitted that Webb fails to disclose this limitation and notes that while Webb discloses the data entry to hold any of a quadword, longword, word, or byte (column 4, lines 66-67), it does not disclose the data bus width of the processor of his invention. In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner referred to remarks made regarding Webb and regarding claim 7. In these remarks, the Examiner asserted that the claim language calls for "a capacity" which is defined as "the ability to receive or contain" by the American Heritage Dictionary, and that the appropriately sized data bus in an ability to receive. Applicants asserted, however, that the Examiner is ignoring the plain language of the claim, which explicitly recites, "a capacity <u>to store</u>..." not just "a capacity". Therefore, the claim is clearly not directed to a capacity <u>to receive</u>, as suggested by the Examiner. Applicants again assert that, as discussed above, *a capacity to store a maximum amount of data that can be written by a store operation* is not defined merely by a data bus size, that the Examiner's remarks regarding a data bus width are irrelevant, and that the limitation discussed above is not taught by Hughes or Webb, as previously admitted by the Examiner. As noted above, the Examiner's Official Notice does not overcome the deficiencies of Hughes and Webb in teaching or suggesting the limitations of claim 4.

For at least the reasons above, the rejection of claim 4 is not supported by the cited art and removal thereof is respectfully requested. Claim 23 includes limitations similar to claim 4, and so the arguments presented above apply with equal force to this claim, as well.

<u>Claims Objected To but Otherwise Allowable</u>:

Claims 6, 7, 17, 25 and 26 were objected to as being dependent upon rejected base claims, but would be allowable if rewritten in independent form. Applicants assert, for at least the reasons presented above regarding the independent claims from which they depend, that these claims are in condition for allowance in their present form.

CONCLUSION

Applicants respectfully submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5500-89400/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: August 20, 2007